Fall 2015

CSCI 6401

Shengru Tu
University of New Orleans

Dr. Shengru Tu Fall 2015

CSCI 6401 Concurrent Programming

Prerequisite: CSCI 4401 - The Principles of Operating Systems

Course Goals:

This course focuses on programming for distributed systems. The class discussion will cover the key problems, theoretical perspectives, solution techniques and typical patterns in distributed programs. Andrews's programming notation is the vehicle to carry out concepts; Java threads and Java sockets will be the tools for implementations. The goal of the lectures is to provide graduate students with a background either for general knowledge or further research used in projects and theses. The emphasis will be on algorithms and implementations.

Students' Learning Outcomes: At the end of this course the students will be able to:

1. command the formal model of concurrent programming – programming logic;
2. analyze the models of different concurrent programming paradigms – remote procedure call, rendezvous, asynchronous message passing, synchronous message passing, and multi-primitive notation for process integration;
3. evaluate the aspect of concurrency of practical software systems, and synthesize solutions based on the classical concurrent algorithms such as heart-beat, prob-echo, token-ring and broadcast algorithms;
4. implement concurrent algorithms in Java and practical communication mechanisms such as sockets and the MapReduce platform.

Texts:

- G. Andrews, "Foundations of Multithreaded, Parallel, and Distributed Programming", 1st edition, Addison-Wesley, 2000 (*required*)
- J. Leskovec, A. Rajaraman and J. Ullman, "Minning of Massive Datasets, Stanford University Infolab", 2014.
- Readings available in the Class Moodle account.

Topics:

Shared-memory concurrent programming
- A programming logic: axioms, interference rules
  - Semantics of concurrency, atomic actions and synchronization
  - Techniques for avoiding interference
  - Safety and liveness properties
- Monitors
- Distributed Programming
- Message passing
  - Networks of filters, CSP, Linda, JavaSpace
- RPC and rendezvous
  - Client and servers, e.g., resource allocation, file server and conversational continuity
- Multi-primitive notation for process integration
  - Interacting parallel processes
  - Prob-echo, heatbeat algorithms

- Broadcast algorithms, e.g., distributed semaphore
- Token-passing algorithms, e.g., distributed mutual exclusion
- Replicated servers

Implementations
- Java thread, Java Sockets
- Other distributed programming platforms such as Java Messaging Service (JMS 2.x)

Algorithms Using MapReduce
- Models of Hadoop and MapReduce
- Distributed algorithms using MapReduce
  - Matrix-vector, matrix-matrix multiplication
  - Distributed algorithms for relational algebra operations
  - Optimization for computation of PageRank

Project

Every student must complete three programming projects: (1) implementation of the communication channel model using Java thread and Java socket technology; (2) a streaming implementation of distributed matrix multiplication; (3) implementation of the distributed semaphore. Details about the project will be given in the third week. Each group should have two students.

Grading:

midterm and final (closed book, in-class): 30% + 35%; projects: 30%

Assignments Policies

Every piece of the work must be produced by the student independently. Copying other people's work will result in an *F* grade in all the involved parties.

Attendance Policy

The class attendance will be checked in every class. A student has to explain the reason if being absent in up to three classes. Missing five or more classes will receive point reduction up to one percent of the course grade for each missing class.

Student Conduct

Be in class on time. Every student is strongly encouraged to ask questions, to participate in class discussions, and respect other students. Sleeping is not permitted in class and it will be treated as an absence.

Academic Integrity

Academic integrity is fundamental to the process of learning and evaluating academic performance. Academic dishonesty will not be tolerated. Academic dishonesty includes, but is not limited to, the following: cheating, plagiarism, tampering with academic records and examinations, falsifying identity, and being an accessory to acts of academic dishonesty. Refer to the Student Code of Conduct for further information. The Code is available online at http://www.studentaffairs.uno.edu.

Students with Special Needs

It is University policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have disabilities that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities should contact the Office of Disability Services as well as their instructors to discuss their individual needs for accommodations. For more information, please go to http://www.ods.uno.edu.